



DECUS

PROGRAM LIBRARY

DECUS NO.	8-187
TITLE	KEYBOARD CONTROLLED BINARY PUNCH
AUTHOR	Edward A. Taft, III
COMPANY	Saint Mark's School Southborough, Massachusetts
DATE	April 7, 1969
SOURCE LANGUAGE	PAL-D

DEC 12

1960



INTRODUCTION

The Keyboard Controlled Binary Punch makes binary tapes of selected areas of core. Its main features are as follows:

1. Keyboard controlled making all operations specified by typed commands.
2. Several blocks of data from separated areas of core can be punched on one tape.
3. Punches field marks for extended memory programming.
4. Occupies only one page of core.
5. Easy to use and especially good for FORTRAN programming. For example, an operating program may be punched onto tape (complete with library routines, operating system, and monitor). Often-used programs saved in this manner may be later run simply by loading the tape with the BIN loader and running them, with no need to fuss with special FORTRAN loaders.

REQUIREMENTS

This program requires a PDP-5, 8, 8/I or 8/S with an ASR-33 and high-speed punch. Extended memory is optional, and greatly increases the convenience of saving large programs.

OPERATION

There are three binary versions of this program available. Version A (LOW) occupies page 1, with a starting address of 0200. Version B (HIGH) occupies page 36, with a starting address of 7400. Version C (SPECIAL) occupies page 37, with a starting address of 7600.

The ASCII version may be reassembled to load into any page by setting SA=the first location on that page.

Load and start the program. It will operate in any field and respond with a CR/LF. Turn on the high-speed punch.

The following commands are now available:

1. L (leader)
Typing the letter L will cause about six inches of BIN leader (code 200) to be punched. The program will then respond auto-

matically with a CR/LF. L also clears the checksum in preparation for punching a program.

2. F n (field)
Typing F, a field number (0-7), and a CR will cause a field mark of the form 3n0 to be punched on the tape. All data thereafter will be taken from the specified field.
3. P mmmm-nnnn (punch)
Typing P, two octal core limits separated by a dash, and CR will cause the specified material to be punched in binary. The data will be taken from the field most recently specified by F, or, if no field has been given, by the field specified by the DATA FIELD register when the program was started.
4. C (checksum)
Typing C will cause the accumulated checksum to be punched, followed by six inches of leader code 2000. The checksum will then be cleared. The program will respond with CR/LF.

All spaces are ignored on input. Any character besides L, F, P, or C typed at the beginning of a line (including CR) will be ignored. No punching will take place, and the program will restart, responding with a CR/LF. Pressing rubout while typing a line will also cause a restart, and the line will be ignored.

No field mark is punched unless an F command is given. Data to be punched is taken from the field indicated by the DATA FIELD register. If it is desired to change the field without punching a field mark, restart the program and set the DATA FIELD register to the desired field.

When punching a tape designed to be loaded into any field, field marks should not be punched, except when punching a tape which must later be loaded into a particular field (or sections of which must be loaded into different fields).

The checksum must be punched immediately after the data. The leader punched after the checksum will cause the BIN loader to compare this checksum to that computed while the tape has been read in.

Enough loader (L) should be punched at the beginning of the tape to make handling easy.

For further information on BIN format, refer to the DEC documentation for the Binary Loader.

Some suggestions for different systems:

1. Without Extended Memory - A free page must be found in which to place the punch program. The SPECIAL version cannot be used for there is no way to load it over the loader. A space between the program and the data is found by running SYMBOLPRINT and can be used to locate the punch routine when saving FORTRAN.

2. Extended Memory - If the SPECIAL version is used, it may be placed in the last page of the field which does not contain the RIM and BIN loaders.

3. 8K Memory and Disk/DEctape Monitor - The monitor head occupies the last page of field 0, and the RIM and BIN loaders are usually placed in field 1. To save the RIM and BIN loaders on the systems device, the following is used: .SAVE BINL ! 17600;

The SPECIAL version may then be loaded into field 1 and used (this may also be saved). To restore the RIM and BIN loaders, call .BINL from the monitor.

4. 8K FORTRAN - After loading the operational program, punch the contents of each field up to and including the page determined by subtracting the number of free pages in that field (printed by the symbol table routine in the loader) from 368. Field marks should be punched on the tape at the beginning of each field.

ERRORS

There is no error detection system. The user should note the following:

1. If the first (non-space; spaces are ignored) character is not L, F, P, or C, it is ignored and the program restarts.

2. Numbers are processed up to the first non-digit. No attempt is made to check for overflow, therefore, a field number not in the range 0-7 may cause severe errors.

3. It is important that the punching strictly follow BIN format, that is, leader must precede data. Several blocks of data may be punched consecutively, but there must be no leader between them. Finally, the checksum must be punched, followed by leader in any amount. The entire sequence may be repeated several times on the same tape, in which case the BIN loader will stop at the end of each section for checksum comparison. The program does not check for illegal sequences.

If a mistake is made, punch leader to clear the checksum and start over.

A typing error may be corrected by pressing rubout and retyping the line if the command has not yet been executed.

/KEYBOARD CONTROLLED BINARY PUNCH

/

/BY EDWARD A. TAFT, III
/ SAINT MARK'S SCHOOL
/ SOUTHBOROUGH, MASS. 01772
/

/FOR 8-FAMILY COMPUTERS, WITH ASR-33 AND HIGH SPEED PUNCH

/

/OCCUPIES ONE PAGE

SA=7600

/

/

	*SA		
7600	1341	START,	TAD C215
7601	4370		JMS COUT
7602	1342		TAD M3
7603	4370		JMS COUT
7604	4350		JMS CIN
7605	1343		TAD CLEAD
7606	7450		SNA
7607	5264		JMP LEADER
7610	1344		TAD CSUM
7611	7450		SNA
7612	5261		JMP CHKSUM
7613	1342		TAD M3
7614	7450		SNA
7615	5242		JMP FLD
7616	1345		TAD CPUN
7617	7640		SZA CLA
7620	5200		JMP START
7621	4317		JMS OCIN
7622	3251		DCA POINT
7623	4317		JMS OCIN
7624	7140		CLL CMA
7625	1251		TAD POINT
7626	3370		DCA COUT
7627	1251		TAD POINT
7630	7120		STL
7631	4303		JMS WOUT
7632	7100		CLL
7633	1651		TAD I POINT
7634	4303		JMS WOUT
7635	2251		ISZ POINT
7636	7700	C7700,	SMA CLA
7637	2370		ISZ COUT
7640	5232		JMP .-6
7641	5200		JMP START
7642	4317	FLD,	JMS OCIN
7643	7104		CLL RAL
7644	7006		RTL
7645	3303		DCA OCNUM
7646	1303		TAD OCNUM
7647	1346		TAD XCDF
7650	3251		DCA .+1
7651	6201	POINT,	CDF
7652	1347		TAD CHECKS
7653	3251		DCA POINT
7654	1303		TAD OCNUM
7655	1236		TAD C7700
7656	4274		JMS POUT
7657	1251		TAD POINT

/PRINT CR
/LF
/GET 1ST CHARACTER
/TEST FOR L,C,F,P
/L=PUNCH 6 INCHES LEADER
/C=PUNCH CHECKSUM AND LEADER
/F=NEW FIELD SETTING
/NONE OF THOSE, IGNORE
/P=PUNCH BINARY DATA
/LOWER LIMIT
/GET UPPER LIMIT
/SET COUNTER
/PUNCH ORIGIN SETTING
/PUNCH THE DATA
/DOES NOTHING HERE
/NEW FIELD, GET FIELD #
/MUST BE 0-7
/CHANGE DATA FIELD
/SAVE CHECKSUM
/FIELD SETTING NOT A PART OF IT
/PUNCH FIELD SETTING
/RESTORE CHECKSUM

7660	5272	JMP	LEADER+6	
7661	1347	CHKSUM,	TAD	CHECKS
7662	7100		CLL	
7663	4303		JMS	WOUT
7664	1236	LEADER,	TAD	C7700
7665	3370		DCA	COUT
7666	1336		TAD	C200
7667	4274		JMS	POUT
7670	2370		ISZ	COUT
7671	5266		JMP.	-3
7672	3347		DCA	CHECKS
7673	5200		JMP	START
		/PUNCH OUTPUT ROUTINE		
7674	0000	POUT,		
7675	6026		PLS	
7676	6021		PSF	
7677	5276		JMP	.-1
7700	1347		TAD	CHECKS
7701	3347		DCA	CHECKS
7702	5674		JMP	I POUT
		OCNUM=.		
		/WORD PUNCH (2 CHARACTERS)		
7703	0000	WOUT,		
7704	3350		DCA	CIN
7705	1350		TAD	CIN
7706	0236		AND	C7700
7707	7012		RTR	
7710	7012		RTR	
7711	7012		RTR	
7712	4274		JMS	POUT
7713	1350		TAD	CIN
7714	0375		AND	C77
7715	4274		JMS	POUT
7716	5703		JMP	I WOUT
		/INPUT AN OCTAL NUMBER		
7717	0000	OCIN,		
7720	3303		DCA	OCNUM
7721	4350		JMS	CIN
7722	1376		TAD	M270
7723	7500		SMA	
7724	5336		JMP	END
7725	1377		TAD	C10
7726	7510		SPA	
7727	5336		JMP	END
7730	3350		DCA	CIN
7731	1303		TAD	OCNUM
7732	7106		CLL	RTL
7733	7004		RAL	
7734	1350		TAD	CIN
7735	5320		JMP	OCIN+1

/PUNCH CHECKSUM

/PUNCH 6 INCHES OF LEADER

/ZERO OUT CHECKSUM

/PUNCH CHARACTER IN AC

/ADD TO CHECKSUM

/GET LEFT HALF

/GET RIGHT HALF

/INITIALIZE TO 0

/GET A CHARACTER

/SEE IF DIGIT

/NO

/NO

/YES, SAVE

/ADD NEW DIGIT

7736	7200	END,	CLA	
7737	1303		TAD	OCNUM
7740	5717		JMP	I OCIN
7741	0215	C215,	215	
7742	7775	M3,	-3	/FOR LF AND "F"
7743	7464	CLEAD,	-314	/FOR "L"
7744	0011	CSUM,	314-303	/FOR "C"
7745	7766	CPUN,	306-320	/FOR "P"
7746	6201	XCDF,	CDF	
7747	0000	CHECKS,	0	/CHECKSUM
		/INPUT A CHARACTER FROM THE TELETYPE		
7750	0000	CIN,	0	
7751	6031		KSF	
7752	5351		JMP	.-1
7753	6036		KRB	
7754	4370		JMS	COUT /ECHO
7755	1365		TAD	M377
7756	7450		SNA	
7757	5200		JMP	START /RUBOUT, RESTART
7760	1366		TAD	TSPC
7761	7450		SNA	
7762	5351		JMP	CIN+1 /IGNORE SPACES
7763	1367		TAD	C240
7764	5750		JMP	I CIN
7765	7401	M377,	-377	
7766	0137	TSPC,	377-240	
7767	0240	C240,	240	
		/OUTPUT A CHARACTER TO THE TELETYPE		
7770	0000	COUT,	0	
7771	6046		TLS	
7772	6041		TSF	
7773	5372		JMP	.-1
7774	5770		JMP	I COUT
		C300=C7700		
7775	0077	C77,	77	
7776	7510	M270,	-270	/DIGIT TESTS
7777	0010	C10,	10	

CHECKS	7747
CHKSUM	7661
CIN	7750
CLEAD	7743
COUT	7770
CPUN	7745
CSUM	7744
C10	7777
C200	7736
C215	7741
C240	7767
C300	7636
C77	7775
C7700	7636
END	7736
FLD	7642
LEADER	7664
M270	7776
M3	7742
M377	7765
OCIN	7717
OCNUM	7703
POINT	7651
POUT	7674
SA	7600
START	7600
TSPC	7766
WOUT	7703
XCDF	7746

/